

17 de Diciembre de 2003

Seminarios PIC



El Proyecto GridWay

Ignacio Martín Llorente
asds.dacya.ucm.es/nacho



Grupo de Arquitectura de Sistemas Distribuidos y Seguridad
Departamento de Arquitectura de Computadores y Automática
Universidad Complutense de Madrid



Laboratorio de Computación Avanzada, Simulación y
Aplicaciones Telemáticas
Centro de Astrobiología CSIC/INTA
Asociado al NASA Astrobiology Institute

1. **Visión Global de la Tecnología Grid**
2. El Grid como Entorno Altamente Dinámico
3. El Proyecto GridWay
4. El Agente de Gestión de Trabajos
5. El Gestor de Rendimiento
6. Comparación con Proyectos Semejantes
7. Trabajo Futuro

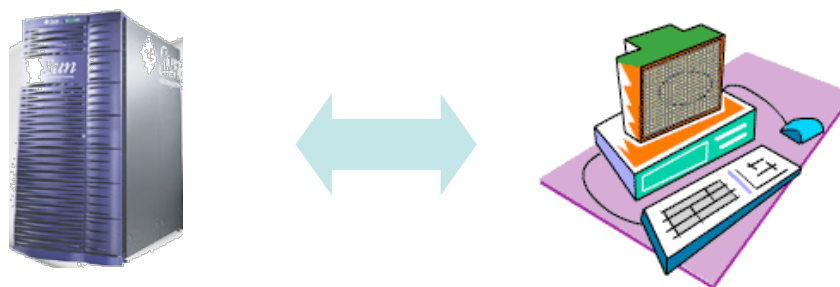
1. Visión Global de la Tecnología Grid

- Supercomputación “Clásica”

Solución centralizada en servidor

- Emulación de terminal del sistema vía red

SUPERCOMPUTADOR



Posibles inconvenientes de la supercomputación basada en servidor

- Falta de escalabilidad
- Equipos muy caros
- Mantenimiento muy caro
- Una vez adquiridos pasan mucho tiempo desaprovechados
- Las demandas de cálculo son puntuales
- Problemas de fiabilidad

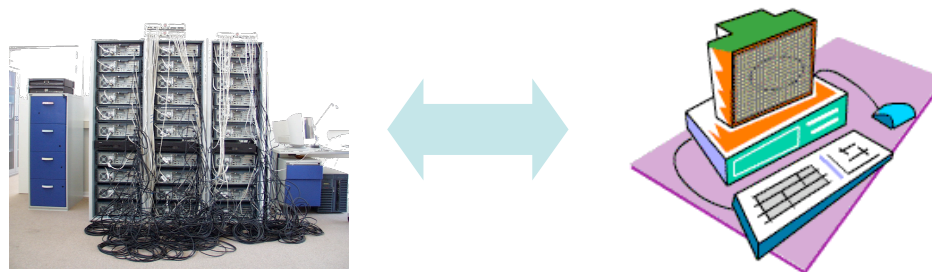
1. Visión Global de la Tecnología Grid

- Computación en Cluster

Solución basada en cluster de equipos

- Cluster de estaciones o computadores personales homogéneo dedicado a computación paralela para obtener una mayor relación prestaciones/coste

CLUSTER



Posibles inconvenientes de la supercomputación basada en clusters

- Coste de las comunicaciones muy alto
 - ✓ Bus lento
 - ✓ Acceso secuencial al bus
 - ✓ Sobrecarga TCP/IP
- Mantenimiento programación

1. Visión Global de la Tecnología Grid

- Aprovechamiento de Recursos en la Intranet

Solución basada en Intranet

- Utilización de los equipos infrautilizados de una red departamental para ejecutar trabajos por medio de una herramienta de gestión de carga



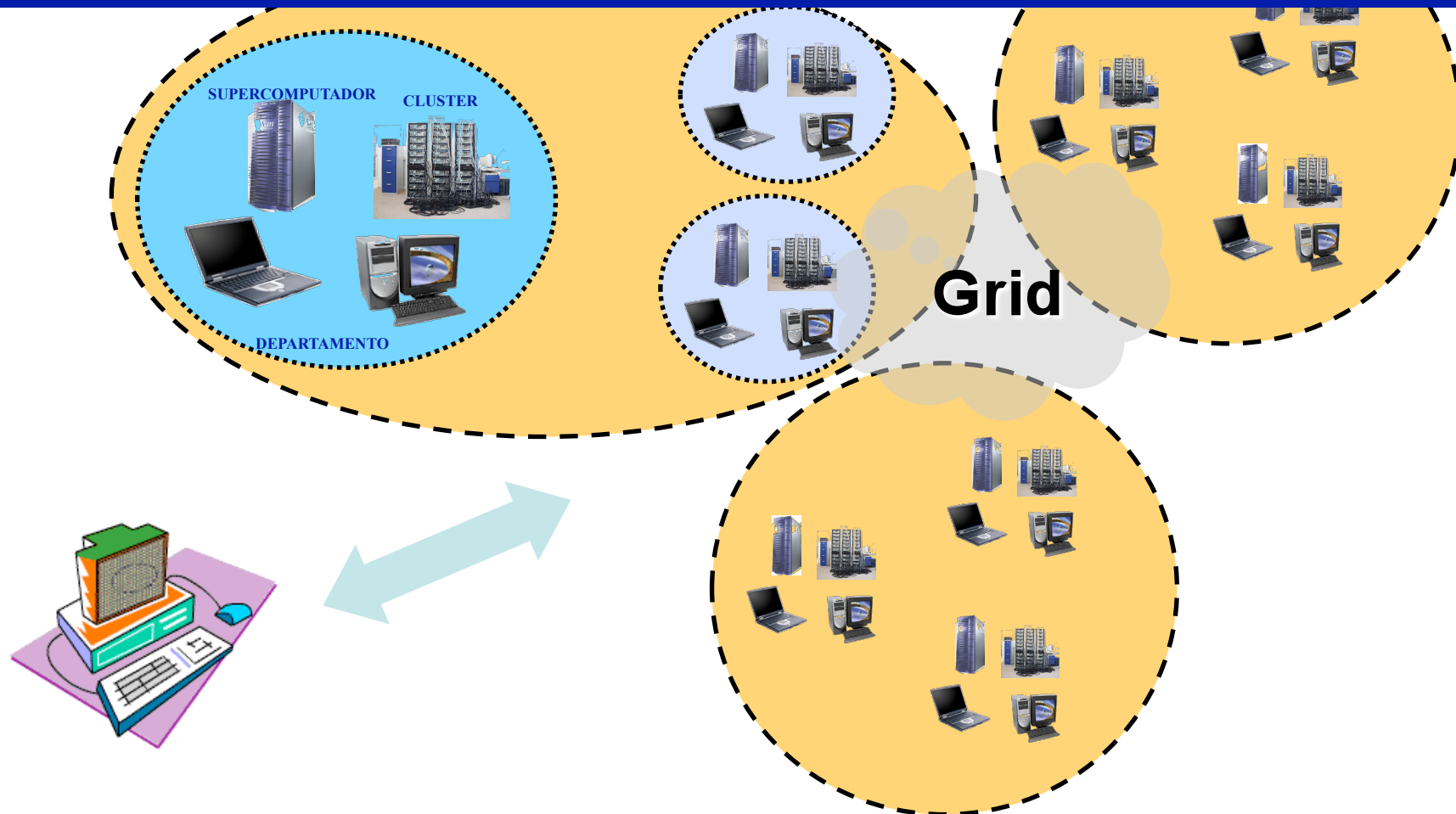
Posibles inconvenientes de la supercomputación basada en clusters

- Escalabilidad limitada a la organización en picos de demanda
- No puedo amortizar mis recursos cuando están desaprovechados
- No puedo compartir recursos con otras organizaciones

1. Visión Global de la Tecnología Grid

- Interconexión de Dominios de Administración

Análoga a las **redes de suministro eléctrico**
Un único punto de acceso a un conjunto de recursos distribuidos geográficamente en diferentes dominios de administración



1. Visión Global de la Tecnología Grid

- La Tecnología Grid

- **Objetivo**
 - Compartir recursos en Internet de forma **uniforme, transparente, segura, eficiente y fiable**
- **Condiciones** (Ian Foster, *What is the Grid? A Three Point Checklist*, 2002)
 - Ausencia de control centralizado
 - Uso de protocolos e interfaces estándar, abiertos y de propósito general
 - Cumplimiento de calidad de servicio
- **La tecnología Grid es complementaria a las anteriores**
 - Interconecta recursos en **diferentes dominios de administración respetando sus políticas internas de seguridad y su software de gestión de recursos en la Intranet**

1. Visión Global de la Tecnología Grid

- Necesidad de un Tecnología de Interconexión de Recursos en Internet

- **Tendencia aproximada**
 - **Incremento lineal del rendimiento** de los computadores más potentes
 - Dos órdenes de magnitud cada 10 años
 - Evolución de **parámetros arquitectónicos**
 - El rendimiento de los procesadores se duplica cada 18 meses, la capacidad de disco cada 12 y el ancho de banda cada 9
- **Requisitos de aplicaciones de gran desafío en computación científica**
 - **Problemas intensivos en computación:** Recursos de cálculo insuficientes para el volumen de datos
 - **Problemas intensivos en datos** Recursos de almacenamiento escasos para el volumen de información
- **Ventajas**
 - **Colaboración y compartición** de recursos en Internet
 - **Alquiler** de recursos
 - **Amortización** de recursos propios

1. Visión Global de la Tecnología Grid

- Capas Grid de Análisis e Investigación



1. Visión Global de la Tecnología Grid

- El Estándar Globus

- **Globus Toolkit**

- Permite compartir recursos localizados en diferentes dominios de administración, con diferentes políticas de seguridad y gestión de recursos

- **Globus es...**

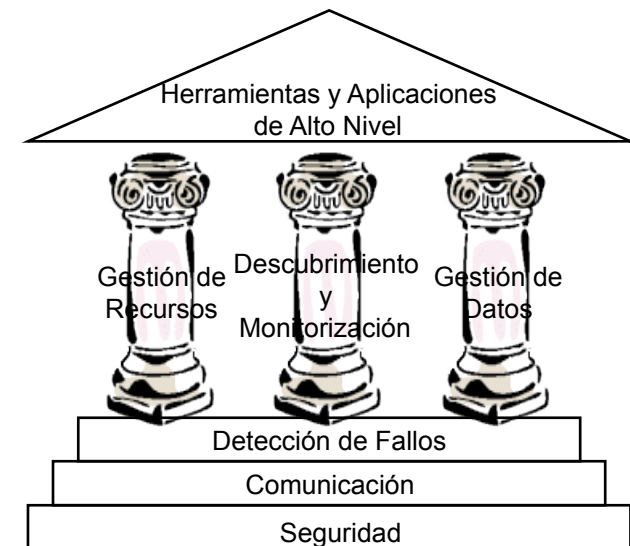
- Un middleware software
- Un conjunto de librerías, servicios y APIs

- **Globus no es...**

- Una herramienta de usuario o planificador
- Una aplicación

- **Componentes Globus**

- Gestión de Recursos (GRAM)
- Gestión de Datos (GridFTP & Replica Catalog)
- Servicio de Información (MDS)
- Infraestructura de Seguridad Grid (GSI)



Contenidos

1. Visión Global de la Tecnología Grid
- 2. El Grid como Entorno Altamente Dinámico**
3. El Proyecto GridWay
4. El Agente de Gestión de Trabajos
5. El Gestor de Rendimiento
6. Comparación con Proyectos Semejantes
7. Trabajo Futuro

2. El Grid como Entorno Altamente Dinámico

- Pasos en la Planificación de un Trabajo



- ¿Dónde ejecuto mi trabajo?
- ¿Qué necesito?
- ¿Cómo envío mi trabajo?
- ¿Cómo va la ejecución?
- ¿Existe un recurso mejor?
- ¿Cómo recupero la salida?

Resource selection

Resource preparation

Job submission

Job monitoring

Job migration

Job termination



Un usuario debe realizar **manualmente todas las acciones** involucradas en la planificación de un trabajo

2. El Grid como Entorno Altamente Dinámico

- Características Dinámicas de un Grid

Cuatro aspectos principales caracterizan un Grid

- Múltiples dominios de administración y autonomía
- Heterogeneidad
- Escalabilidad
- Adaptabilidad

Alta Tasa de Fallos

- Red
- Recurso

Precio Dinámico de Recursos

- Hora del día
- Carga del recurso

Disponibilidad Dinámica de Recursos

- Cancelación trabajos
- Adición/exclusión recursos

Grid



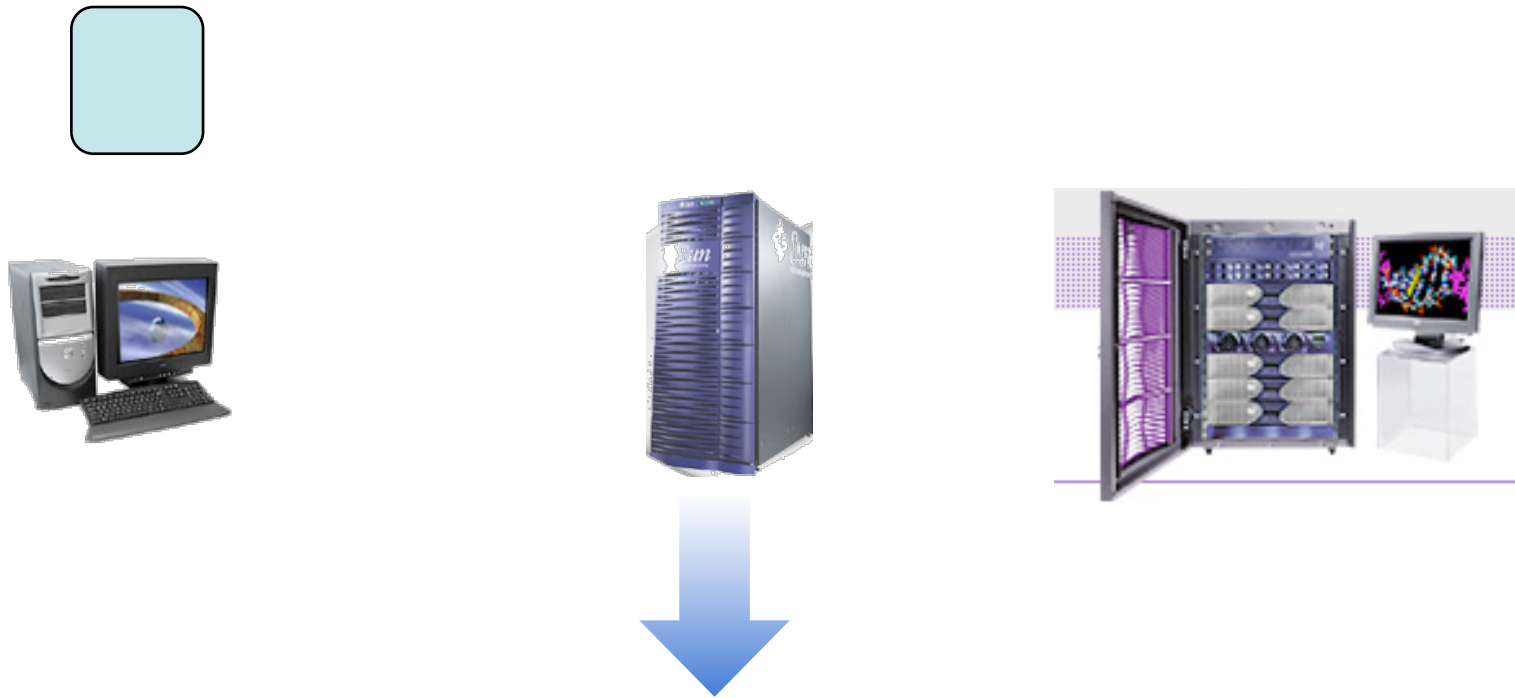
Carga Dinámica de Recursos

- Recursos compartidos
- Recursos ociosos se saturan

Un trabajo debe poder **migrar entre los recursos de un Grid** para obtener un nivel suficiente de rendimiento y tolerancia a fallos.

2. El Grid como Entorno Altamente Dinámico

- Aplicaciones Auto-Adaptativas



Nueva generación de aplicaciones que se basa en la **capacidad para adaptar su ejecución** a sus propios requisitos dinámicos

2. El Grid como Entorno Altamente Dinámico

- Eventos Posibles de Re-planificación

- **Originados por el Grid**

- Se descubre un **nuevo recurso** con una clasificación mayor que el actual (migración oportunista)
- El recurso remoto o su conexión de red **fallan**
- **El trabajo es cancelado o suspendido** por el administrador del recurso

- **Originados por la aplicación**

- La aplicación detecta un **deterioro del rendimiento** o una violación de contrato de rendimiento
- **La aplicación varía sus requisitos** de forma que el nuevo recurso no los satisface (auto-migración)

Contenidos

1. Visión Global de la Tecnología Grid
2. El Grid como Entorno Altamente Dinámico
- 3. El Proyecto GridWay**
4. El Agente de Gestión de Trabajos
5. El Gestor de Rendimiento
6. Comparación con Proyectos Semejantes
7. Trabajo Futuro

3. El Proyecto GridWay - Objetivo Principal

Investigar sobre el envío, la planificación y la gestión de trabajos en un entorno tan complejo como el Grid

Desarrollar una tecnología que, usando los servicios básicos ofrecidos por Globus, permita a los usuarios lanzar trabajos y **olvidarse de los detalles de implementación y las características dinámicas de un Grid**



Directrices de Diseño

- Fácilmente **Adaptable** (diseño modular)
- Fácilmente **Escalable** (arquitectura descentralizada)
- Fácilmente **Instalable** (usuario, servicios estándar)
- Fácilmente **Aplicable** (conjunto amplio de aplicaciones)

3. El Proyecto GridWay

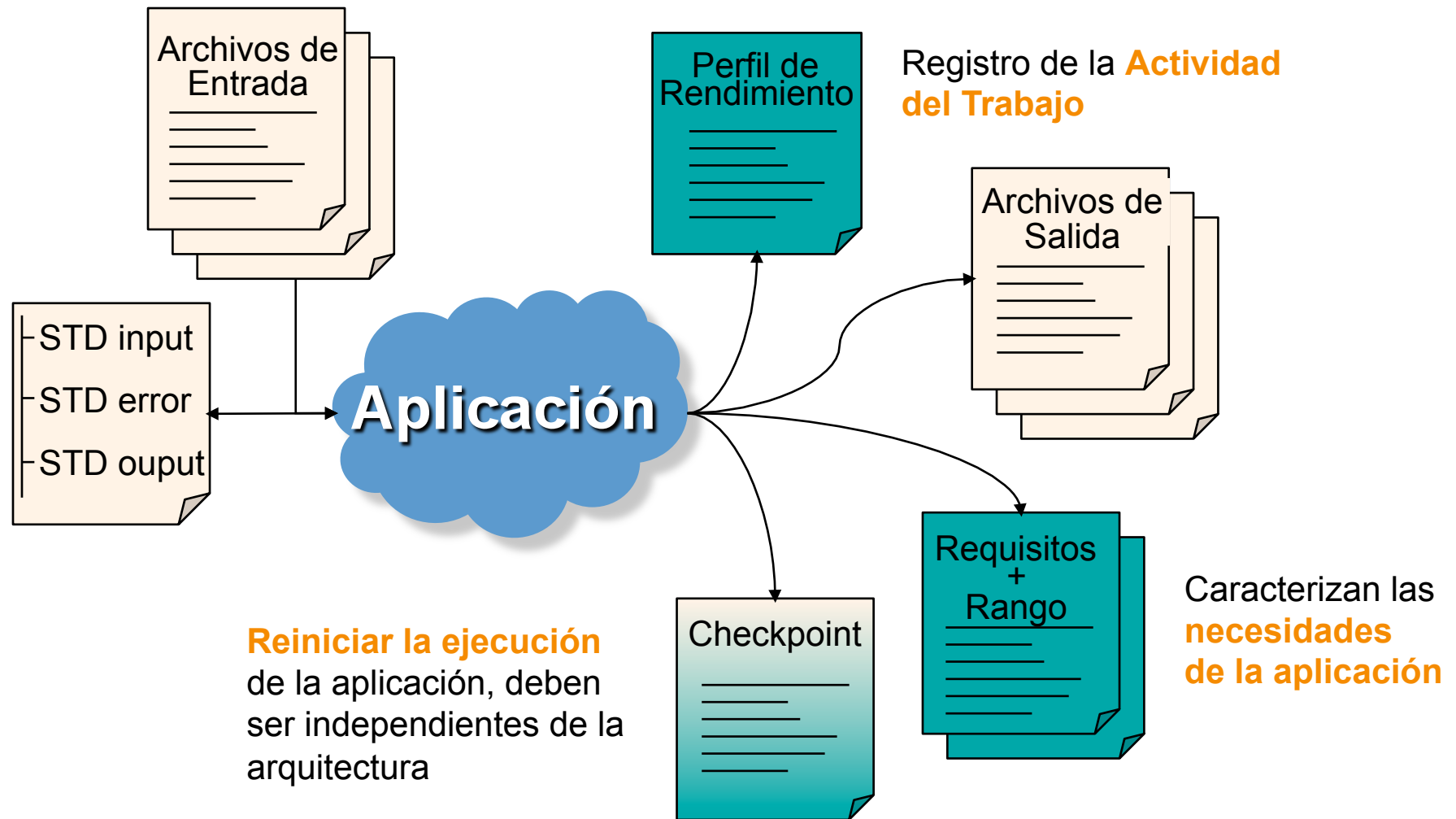
- Objetivo Adicional

- **Desafíos del Despliegue de Tecnología Grid**
 - **Desafíos Técnicos**
 - **Desafíos Sociopolíticos**
- **Objetivo**
 - Investigar sobre alternativas para que los administradores puedan limitar el uso de sus recursos compartidos, esto es **gestión del rendimiento local** de los recursos en un entorno Grid
- **Resultado**
 - **La Herramienta GRPM:** Componentes que permitan decidir a los administradores la cantidad de recursos que desean dedicar al Grid

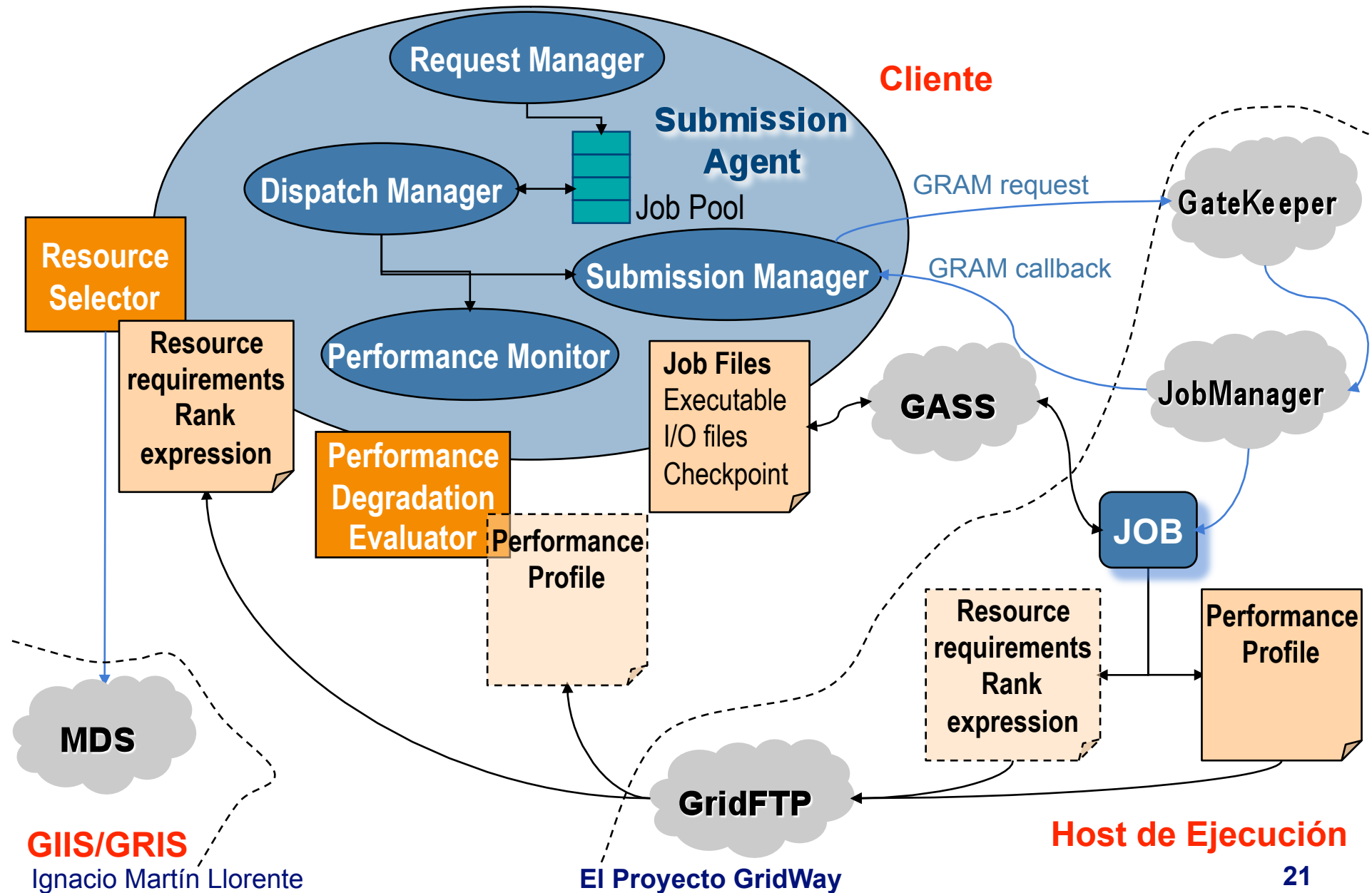
Contenidos

1. Visión Global de la Tecnología Grid
2. El Grid como Entorno Altamente Dinámico
3. El Proyecto GridWay
- 4. El Agente de Gestión de Trabajos**
5. El Gestor de Rendimiento
6. Comparación con Proyectos Semejantes
7. Trabajo Futuro

4. El Agente de Gestión de Trabajos - Modelo Extendido de Aplicación

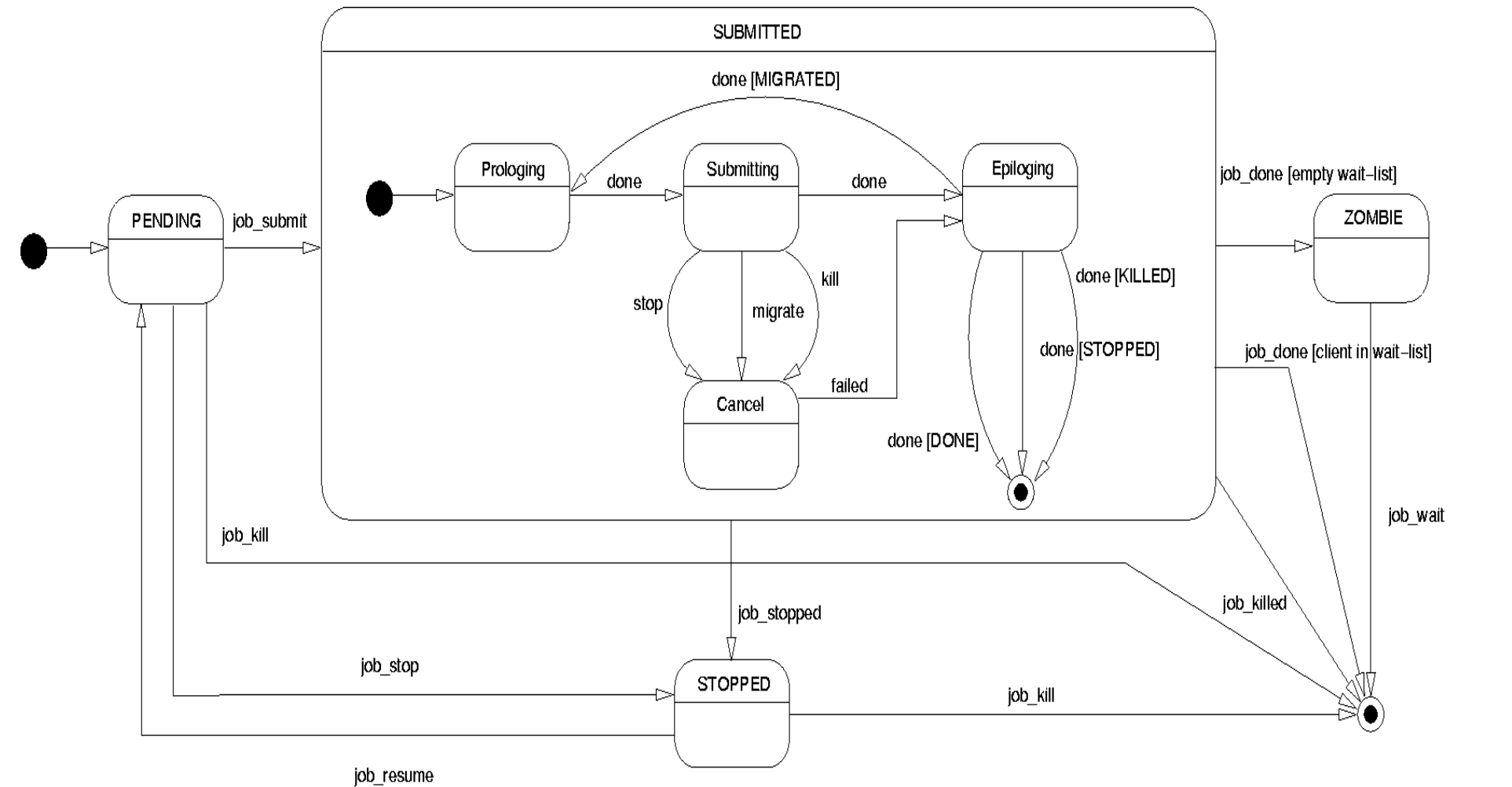


4. El Agente de Gestión de Trabajos - Arquitectura

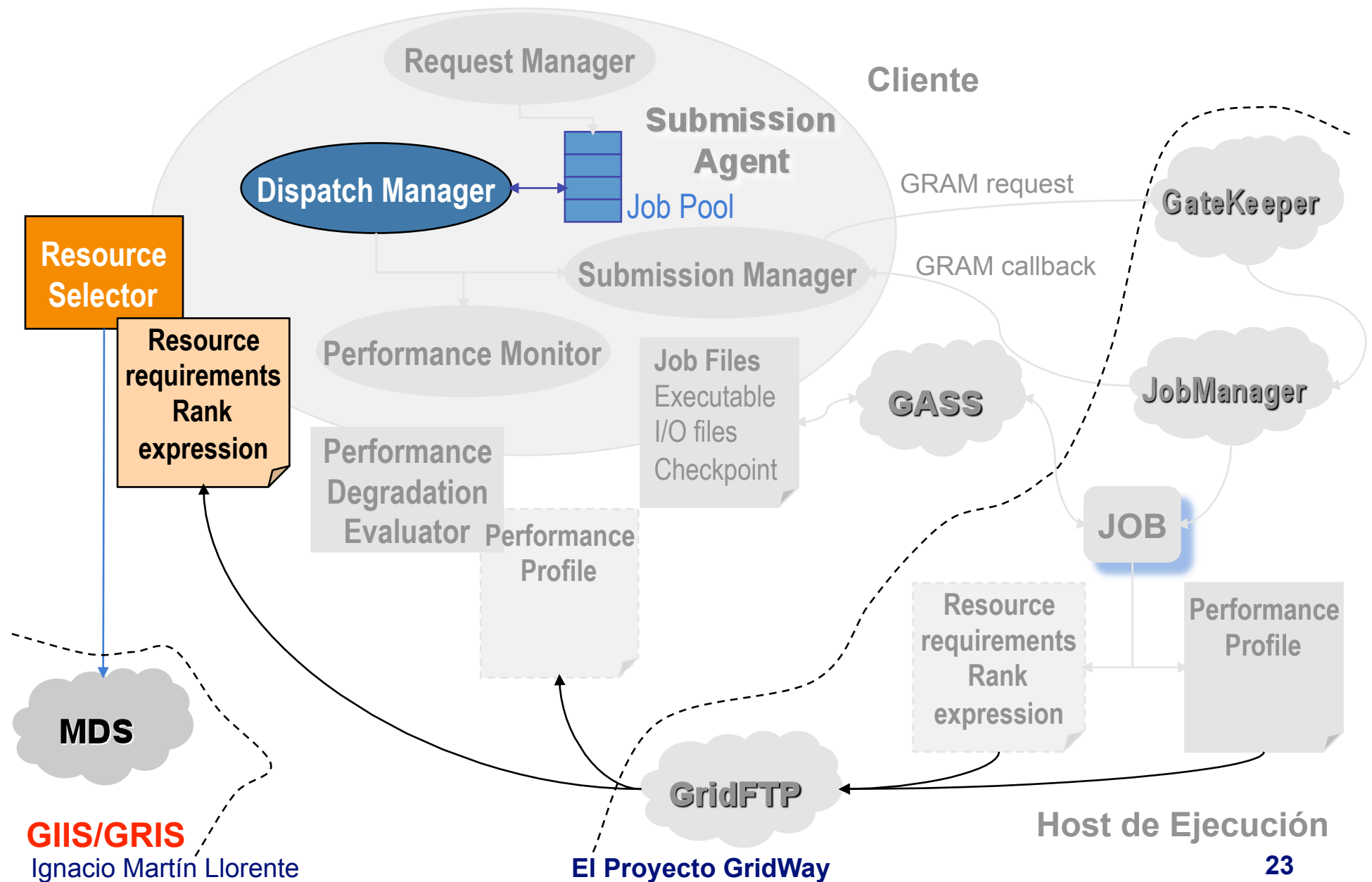


4. El Agente de Gestión de Trabajos

- Estados de un Trabajo



4. El Agente de Gestión de Trabajos - Planificación de Trabajos



4. El Agente de Gestión de Trabajos

- Descubrimiento y Selección de Recursos

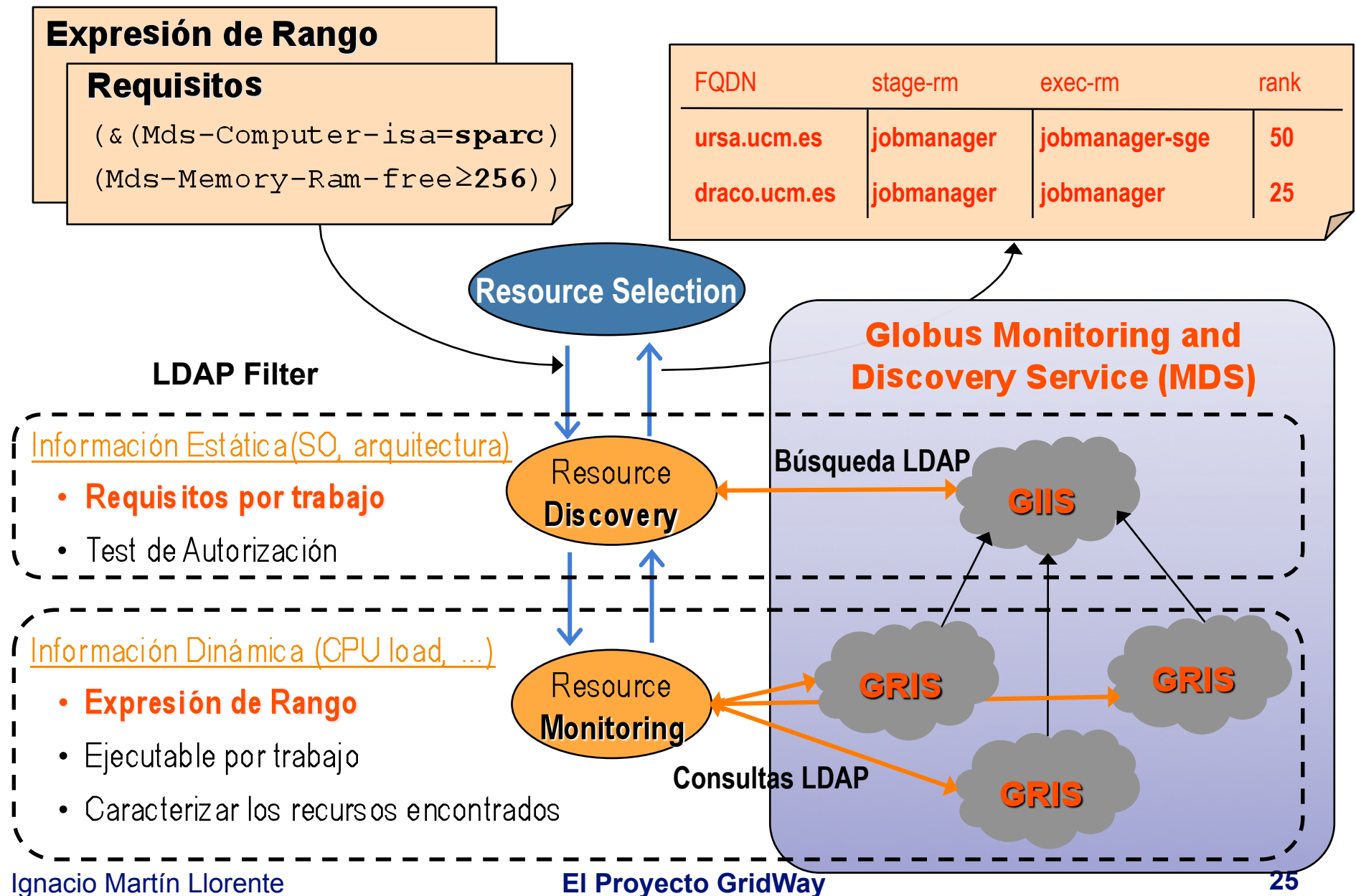
Dada la naturaleza **heterogénea** y **dinámica** de un Grid deben establecerse

- **Requisitos** que deben satisfacer los posibles recursos
 - Sistema operativo, arquitectura, software específico,...
 - Posibilidad de usar el recurso: Autorización, disponibilidad,...
- Criterio (**rango**) para cualificar los recursos factibles
 - **Modelo de Rendimiento**, en términos de métricas específicas de la aplicación
 - Actualización dinámica de rangos

Información estática y dinámica obtenida de los servicios de información del Grid

- Lista predefinida de recursos y Scripts de Prueba (`uptime`, `pbsnodes`, ...)
- Globus MDS
- Network Weather Service
- Replica Location Service

4. El Agente de Gestión de Trabajos - Descubrimiento y Selección de Recursos

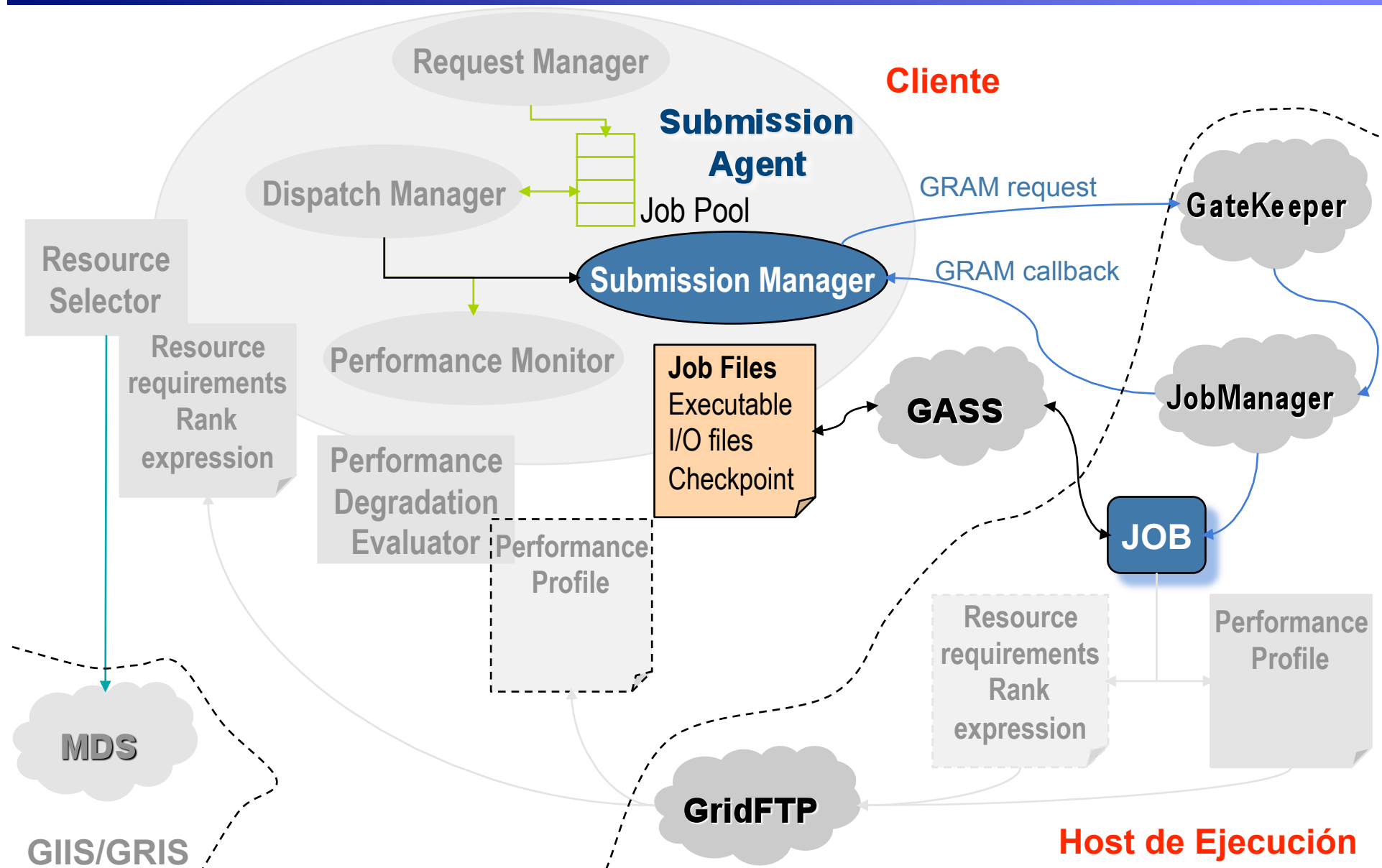


4. El Agente de Gestión de Trabajos

- Planificación de Trabajos en Grids

- **Objetivo del Planificador:** Optimizar el rendimiento en función de una métrica adecuada.
 - Productividad del sistema (meta-planificadores, o multi-planificadores)
 - Optimización del uso de un recurso
 - **Rendimiento de la Aplicación** (*Application Level Schedulers*)
 - **Aplicaciones Tradicionales**
 - **Aplicaciones de Alto Rendimiento:** cuyos requisitos computacionales únicamente puede satisfacerse mediante la unión de múltiples supercomputadores (**MPICH-G2**, **GrADS**)
 - **Aplicaciones de Alta Productividad:** Consisten en múltiples ejecuciones sobre un espacio de parámetros (**Condor/G**, **AppleS**, **Nimrod/G**)
 - **Aplicaciones Auto-Adaptativas:** Capaces de tomar decisiones sobre la selección de recursos, para satisfacer sus requisitos dinámicos (**Cactus**) o QoS de usuario (**Nimrod/G**)
- La planificación de forma implícita o explícita esta basada en una predicción de rendimiento (modelo). **Planificación Adaptativa**, usa información dinámica sobre el estado de los recursos del Grid.

4. El Agente de Gestión de Trabajos - Envío de Trabajos



4. El Agente de Gestión de Trabajos - Transferencia de Ficheros

Transferencia de Archivos y Recuperación de la Salida (Estáticos):

- **Prólogo:** Creación del directorio de experimento remoto (diferentes usuarios) y transferencia de ficheros.
- Wrapper: Ejecución del Trabajo.
- **Epílogo:** Transferencia de ficheros de salida y eliminación del directorio remoto.

La transferencia de ficheros se realiza siempre usando el ***jobmanager fork*** mediante un **modelo de transferencia inverso (HTB, High Throuput Broker)**.

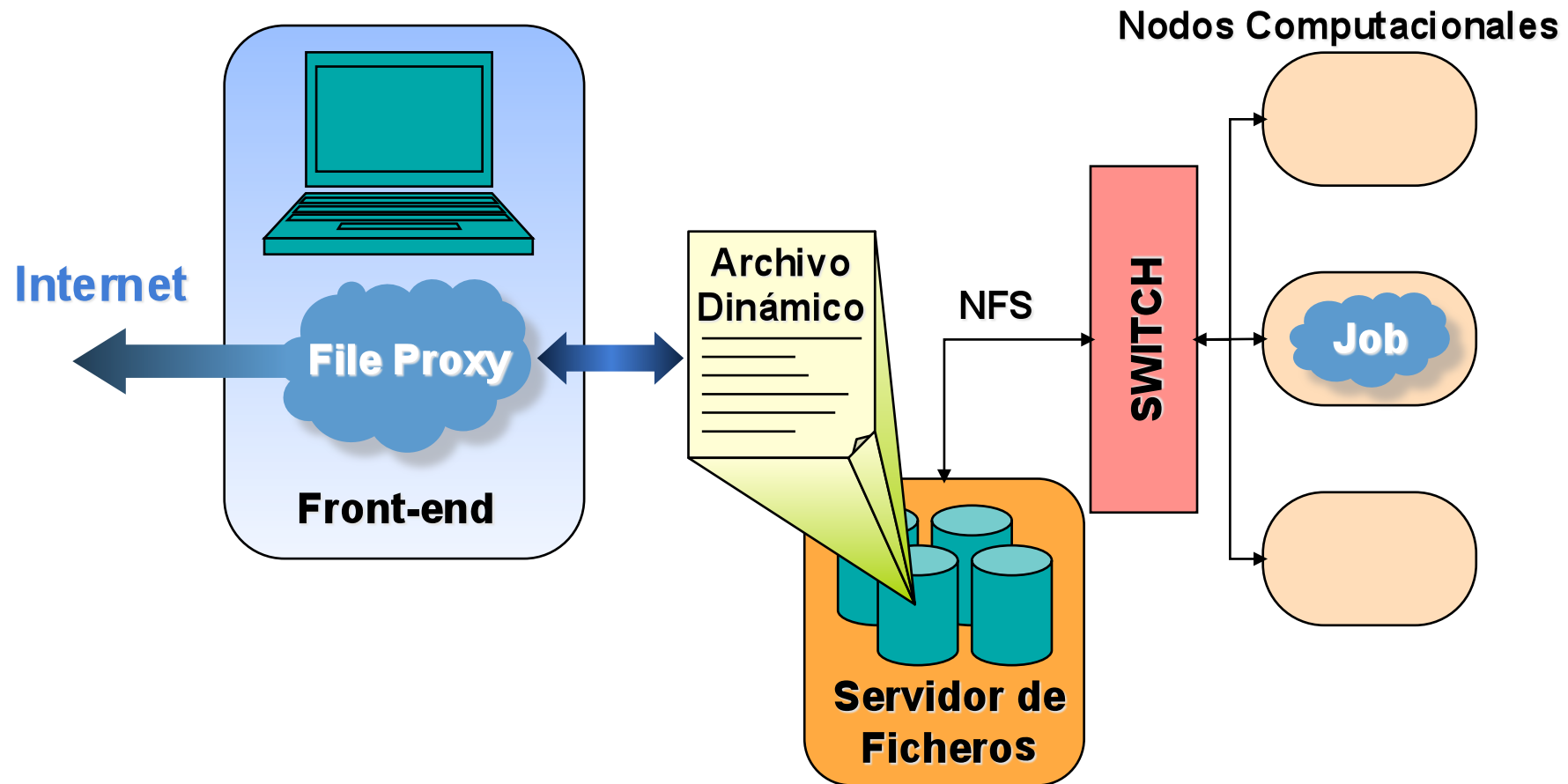
Transferencia y Ejecución en un paso (Nimrod/G, EDG):

- Ajuste de los parámetros del RSL (**maxtime**)
- Compatibilidad con diferentes *jobmanagers* para la ejecución (**Condor-standard**)
- Sistemas cerrados

Estrategias:

- **Directa**, dónde los archivos se almacenan en el cliente.
- Gestión de **Réplicas** y **transferencia entre terceros**.
- Acceso a **bases de datos distribuidas** (*protein data bank*)
- Uso de **GASS-cache** (trabajos paramétricos)
- **Compresión de archivos**

4. El Agente de Gestión de Trabajos - Transferencia de Ficheros



Transferencia de Archivos Dinámicos: Archivos generados por el trabajo durante su ejecución (rango, requisitos y perfil de rendimiento)

4. El Agente de Gestión de Trabajos - Transferencia de Ficheros

Gestión de Fallos:

- Salida estándar de los scripts
- Número de reintentos predeterminado por trabajo (Re-planificación, Parada)
- **Saturación** del sistema de ficheros remoto (globus-gass-cache-destroy)

Parámetros del *job-template*

```
ON_FAILURE          = reschedule
NUMBER_OF_RETRIES   = 5
# Executable Parameters
EXECUTABLE_FILE      = NS3D.${GW_ARCH}
EXECUTABLE_ARGUMENTS = input
INPUT_FILES          = "grid32x32x32.${GW_ARCH}  grid32x32x32, input"
OUTPUT_FILES         = "ns3dout.residual  ns3dout.residual.${GW_JOB_ID},
                        out.grid  out.grid.${GW_JOB_ID}"

# Standard Files
STDIN_FILE           = /dev/null
STDOUT_FILE          = stdout_file.${GW_JOB_ID}
STDERR_FILE          = stderr_file.${GW_JOB_ID}
# Migration related files
RESTART_FILES        = checkpoint.ascii
```

4. El Agente de Gestión de Trabajos

- Envío del Trabajo

Wrapper: Realiza el **envío del trabajo** (RSL) y captura el **código de salida**:

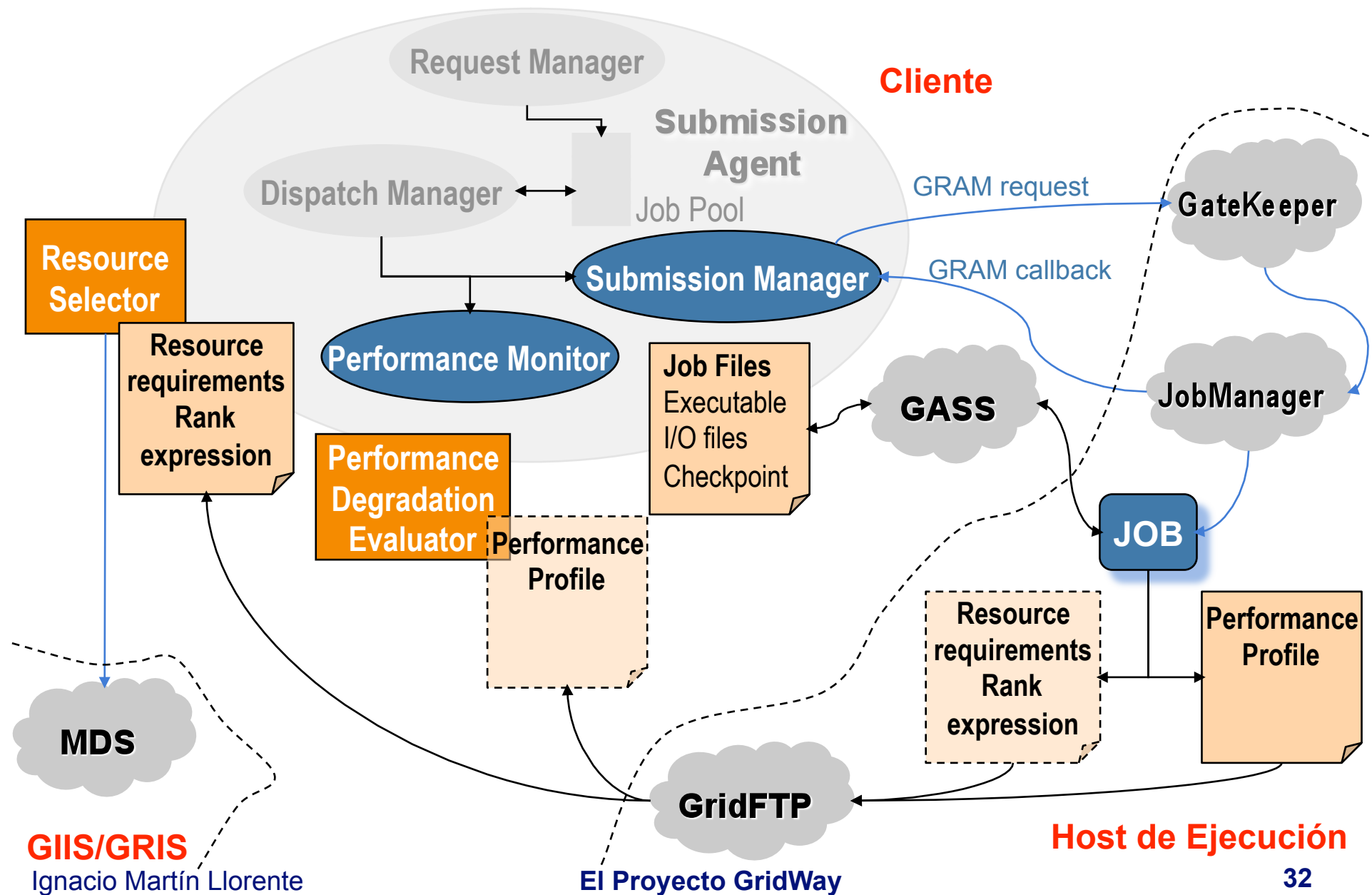
- El código de salida es **cero**, el trabajo termina exitosamente.
- El código de salida es **distinto de cero**, el trabajo termina.
- El código de salida **no ha sido fijado**, el trabajo ha sido cancelado y se re-planifica

Globus no ofrece ningún mecanismo para recuperar el código de salida de un trabajo

Definición de Trabajos Complejos



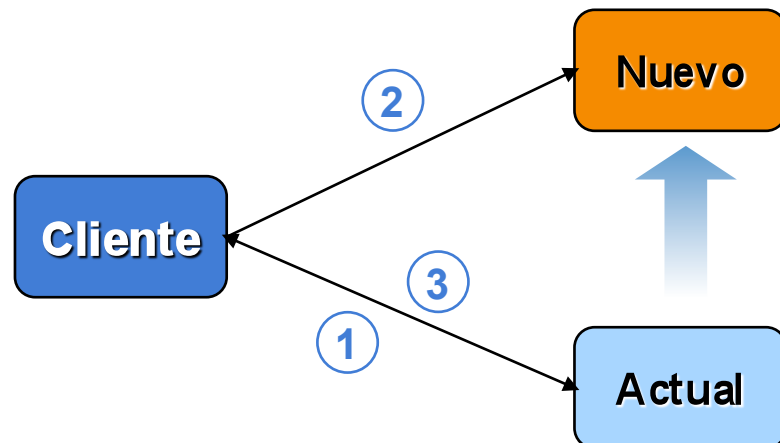
4. El Agente de Gestión de Trabajos - Migración y Monitorización



4. El Agente de Gestión de Trabajos - Eventos de Monitorización

La **adaptación de un trabajo** se realiza mediante su migración automática cuando:

- Se descubre un **recurso mejor** (migración oportunista).
 - El recurso remoto o su conexión de red **fallan**.
 - El trabajo es **cancelado**, o permanece **suspendido (PENDING)** demasiado tiempo.
 - Se detecta una **degradación de rendimiento**.
 - Los requisitos de la aplicación cambian (**auto-migración**)
- Grid
- Aplicación



1. **Cancelación** del trabajo
2. Envío del **Prólogo** al nuevo host, transferencia de ficheros
3. Envío del **Epílogo** al host actual, copia de los ficheros de checkpoint (fallo)

4. El Agente de Gestión de Trabajos

- Características de la Monitorización

Factores que influyen en la **migración de un trabajo**:

- Fracción de **trabajo realizado** por la aplicación (**GrADS**)
- **Proximidad del recurso** al que se migrará
- **Métrica** que estime la ganancia obtenida debida a la migración (**GrADS**)

Migration Gain (migración oportunista y degradación de rendimiento):

$$G_m = \frac{Rank(h_{n-1}, t_{n-1}) - Rank(h_n, t_n)}{Rank(h_{n-1}, t_{n-1})} \geq \text{Umbral de usuario}$$

Políticas de Re-planificación

- Petición de **re-planificación** por parte de **usuario** (**migración automática**)
- Cada **discovering interval** para descubrir un nuevo recurso (**evaluación G_m**)
- El *Performance Monitor* detecta un **degradación de rendimiento**
 - Mediante el Performance Degradation Evaluator (**evaluación G_m**)
 - Tiempo máximo de suspensión (**migración automática**)
- El *submission Manager* detecta un fallo (cancelación, o conexión) (**evaluación G_m**)

4. El Agente de Gestión de Trabajos

- Monitorización

Monitorización de la Ejecución de un Trabajo:

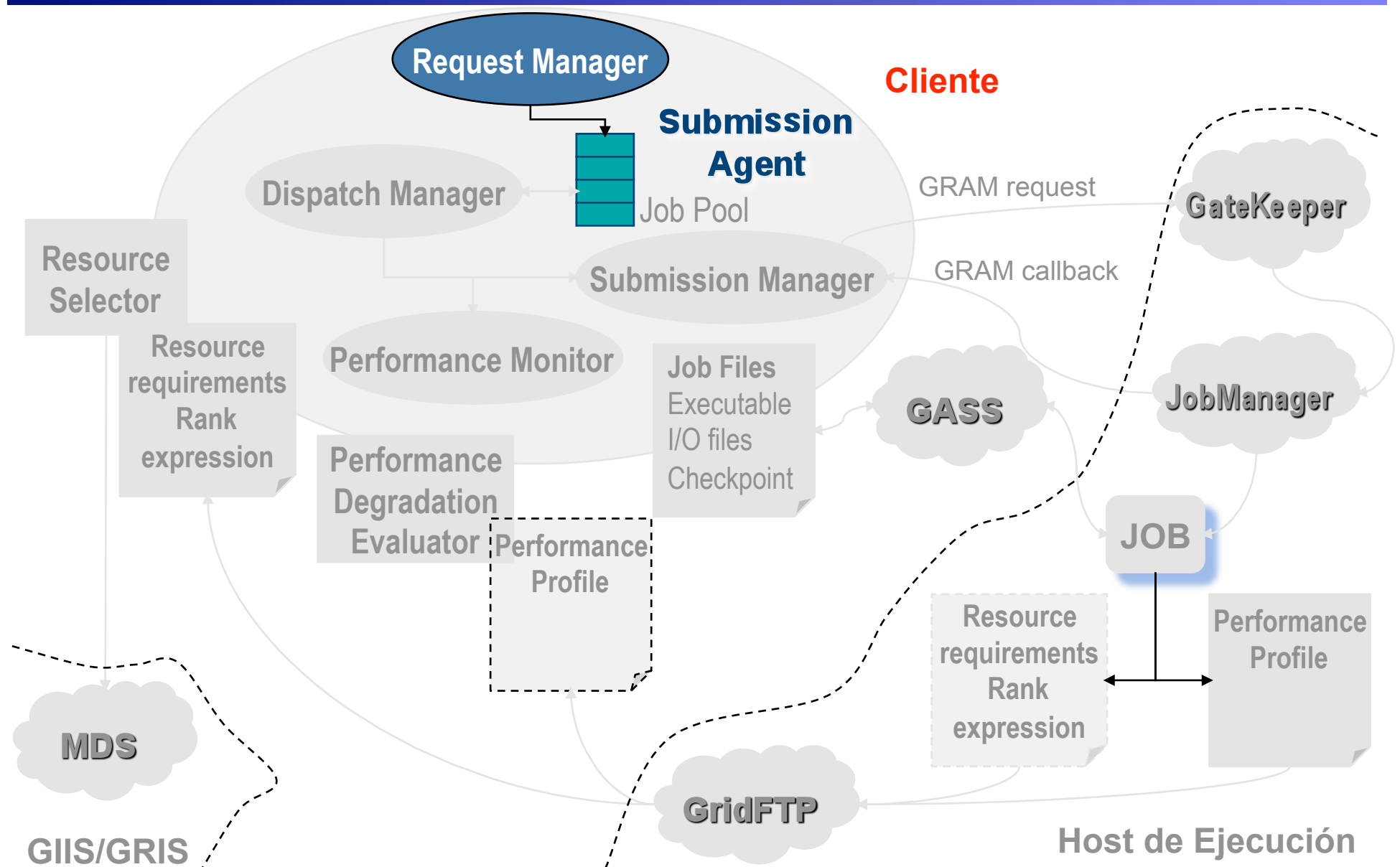
- Actualizando los **callbacks** del protocolo **GRAM de Globus**, cuando el envío de un trabajo falla se recibe `GLOBUS_GRAM_PROTOCOL_JOB_STATE_FAILED`.
- **Comprobando periódicamente** el **Jobmanager** del trabajo. Si se produce un fallo se comprueba el **Gatekeeper**, si el Gatekeeper responde se reinicia un Jobmanager que vigile el trabajo. En caso contrario se re-planifica el trabajo. (**Condor/G**)
- Mediante el **código de salida** del trabajo.
- El rendimiento del trabajo se monitoriza mediante el **perfil de rendimiento** del trabajo, en términos de métricas específicas de la aplicación.
- **Tiempo de suspensión** del trabajo
- Análisis de la **salida estándar en las fases de transferencia** de ficheros para detectar fallos de ejecución (**Nimrod/G**)

4. El Agente de Gestión de Trabajos - Monitorización y Migración

Parámetros del *job-template*

- Intervalo de encuesta del jobmanager y Gatekeeper
`POLL_TIMEOUT` = 30
- Máximo tiempo de suspensión del trabajo en el sistema de colas remoto
`MAX_SUSPENSION_TIME` = 360
- Determinación del intervalo de búsqueda de nuevos recursos
`MAX_DISCOVERY_TIME` = 120
- Intervalo de comprobación del rendimiento de la aplicación
`PERFORMANCE_CHECK_TIME` = 600

4. El Agente de Gestión de Trabajos - Interfaz de Usuario



4. El Agente de Gestión de Trabajos - Interfaz de Usuario

Interfaz de Usuario:

- **gwps:** Muestra información y estado de cada trabajo

<u>JID</u>	<u>AID</u>	<u>TID</u>	<u>DM</u>	<u>SM</u>	<u>GSM</u>	<u>STIME</u>	<u>ETIME</u>	<u>EXETIME</u>	<u>EXIT</u>	<u>HOST</u>	<u>TEMPLATE</u>
0	--	--	submitted	prologue	--	--:--	--:--	--:--	--	columba	job_template
1	--	--	zombie	done	--	27:37	28:07	00:30	0	ursa	job_template
7	--	--	pending	done	--	--:--	--:--	--:--	--	draco	job_template

- **gwhistory:** Muestra la historia de ejecución

<u>HOST</u>	<u>RANK</u>	<u>STIME</u>	<u>ETIME</u>	<u>EXETIME</u>	<u>MIGRATION</u>	<u>REASON</u>
columba.dacya.ucm.es	100	--:--	--:--	--:--	--	
ursa.dacya.ucm.es/jobmanager-grd	50		27:41	27:52	00:11	discovery timeout

- **gckill:** Envío de señales a un trabajo (kill, stop, resume, reschedule)
- **gws submit:** Envío de un trabajo, o trabajo en array
- **gwwait:** Espera el estado zombie de un trabajo (any, all, set)

API de Cliente: Permite la interacción con cada módulo, (**subconjunto DMRAA**)

4. El Agente de Gestión de Trabajos - Resultados más Relevantes

• PLANIFICACIÓN ADAPTATIVA DE TRABAJOS

Descripción del CAB/UCM Grid:

Host	Modelo	Hz	S.O.	Memoria	Nodos	GRAM
<i>ursa</i>	Sun Blade 100	500Mhz	Solaris 8	256MB	1	fork
<i>draco</i>	Sun Ultra 1	167Mhz	Solaris 8	128MB	1	fork
<i>pegasus</i>	Intel Pentium 4	2.4 Ghz	Linux 2.4	1GB	1	fork
<i>babieca</i>	Alpha DS10	466 Mhz	Linux 2.4	1GB	4	pbs
<i>solea</i>	Sun Enterprise 250	296 Mhz	Solaris 8	256MB	2	fork

Descripción del Experimento:

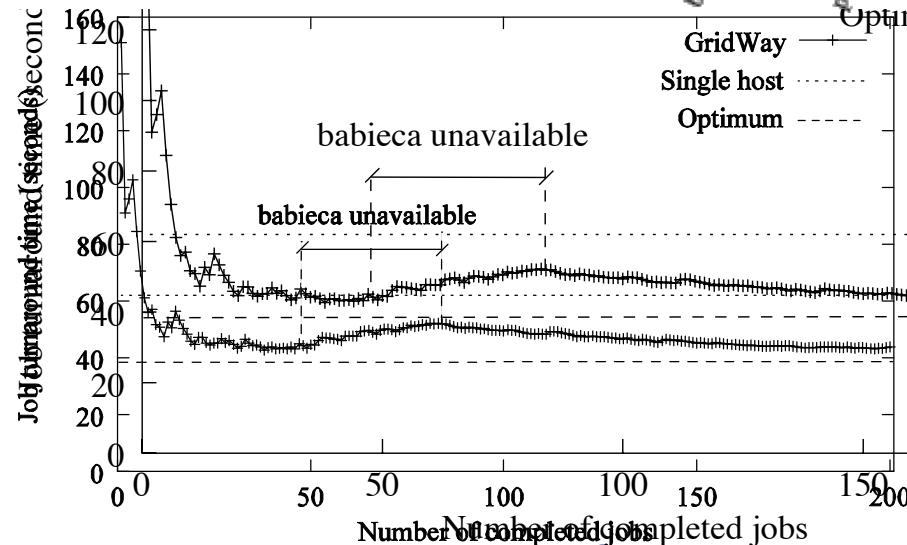
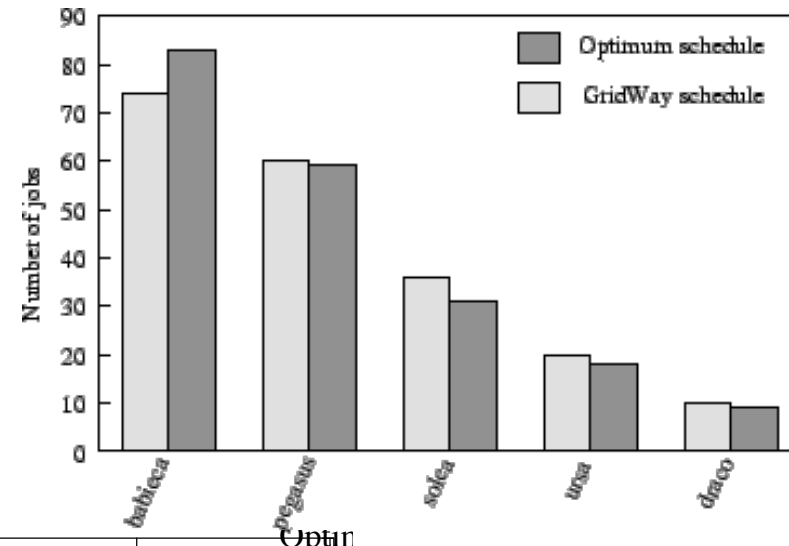
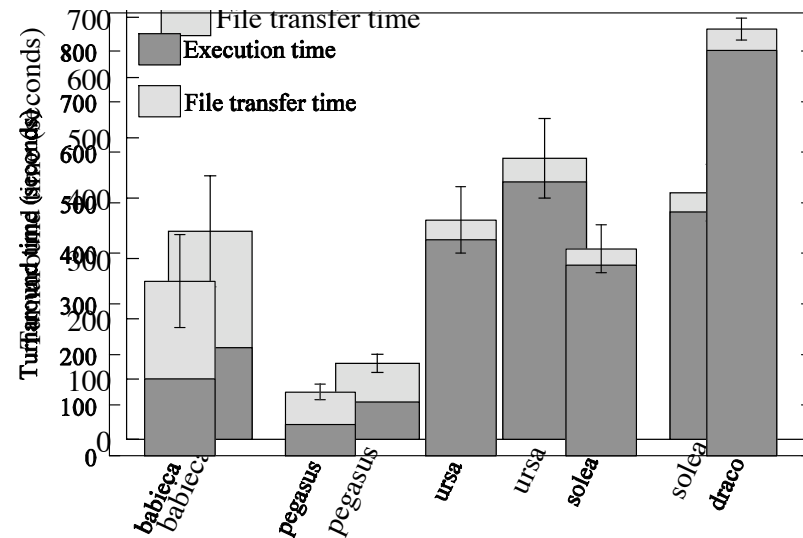
- Ejecución de una aplicación de Alta Productividad (CFD $Re=10^2 - 10^4$)
- 200 tareas independientes



4. El Agente de Gestión de Trabajos - Resultados más Relevantes

• PLANIFICACIÓN ADAPTATIVA DE TRABAJOS

Resultados del Experimento:



4. El Agente de Gestión de Trabajos - Resultados más Relevantes

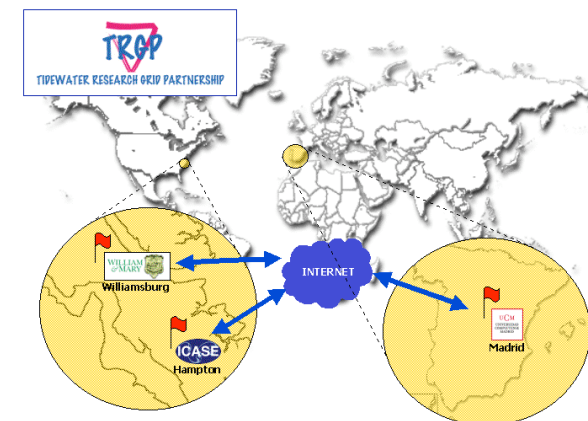
• EJECUCIÓN ADAPTATIVA DE TRABAJOS

Descripción del TRGP Grid:

Nombre	Modelo	Rend.	Nodos	SO	Memoria	GRAM
<i>sciclone</i>	Sun UltraSPARC	115Gflops	160	Solaris 8	54GB	PBS
<i>coral</i>	Intel Pentium	89Gflops	103	Linux 2.4	56GB	PBS
<i>whale</i>	Sun UltraSPARC	1.8Gflops	2	Solaris 7	4GB	fork
<i>urchin</i>	Sun UltraSPARC	672Mflops	2	Solaris 7	1GB	fork
<i>carp</i>	Sun UltraSPARC	900Mflops	1	Solaris 7	256MB	fork
<i>tetra</i>	Sun UltraSPARC	800Mflops	1	Solaris 7	256MB	fork
<i>bonito</i>	Sun UltraSPARC	720Mflops	1	Solaris 7	256MB	fork

Descripción del Experimento:

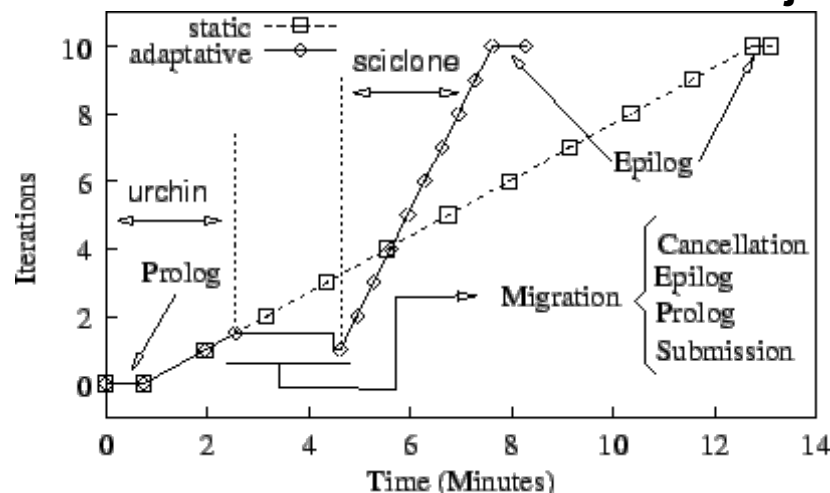
- Ejecución de una única aplicación CFD
 - ✓ Descubrimiento de un Recurso Mejor
 - ✓ Detección de una Degradación del Rendimiento
 - ✓ Detección de un Fallo
 - ✓ Superación del Tiempo Máximo de Suspensión
 - ✓ Cambio de Requisitos



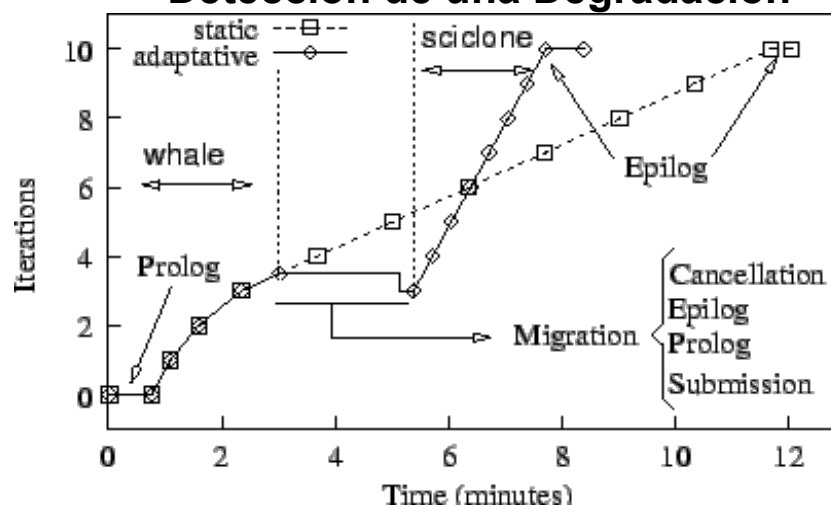
4. El Agente de Gestión de Trabajos - Resultados más Relevantes

• EJECUCIÓN ADAPTATIVA DE TRABAJOS

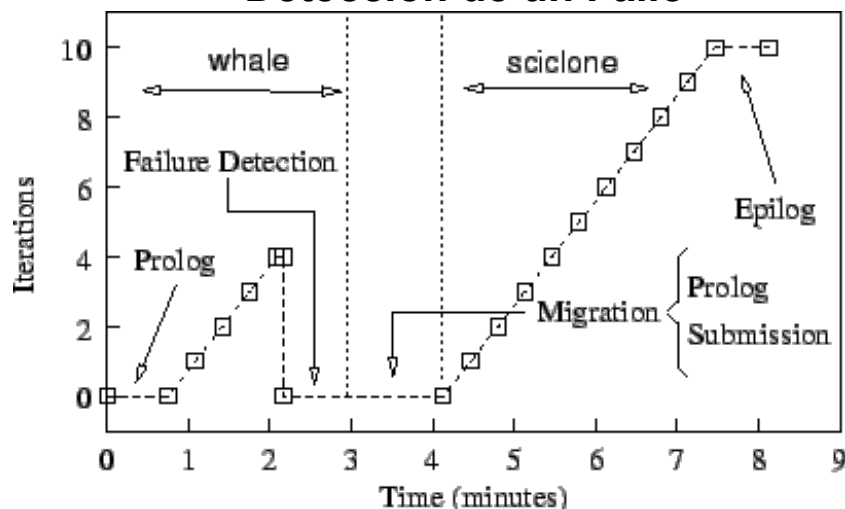
Descubrimiento de un Recurso Mejor



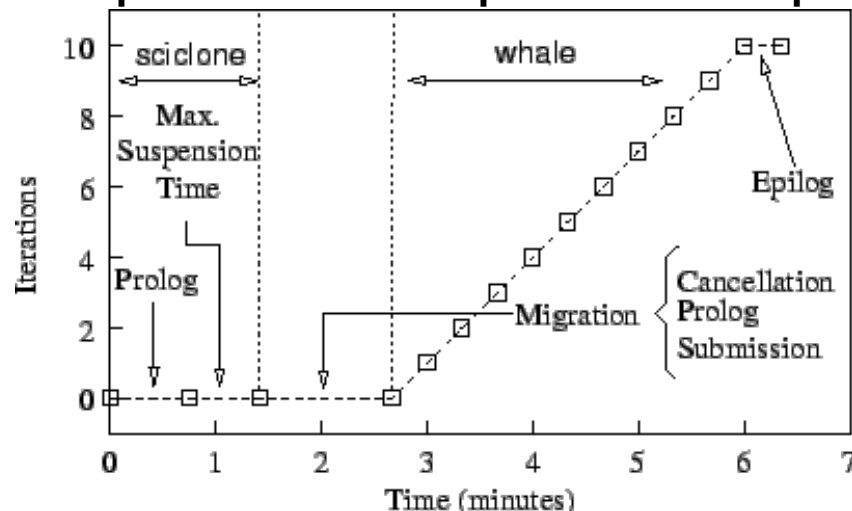
Detección de una Degradación



Detección de un Fallo



Superación del Tiempo Máximo Suspensión



4. El Agente de Gestión de Trabajos - Resultados más Relevantes

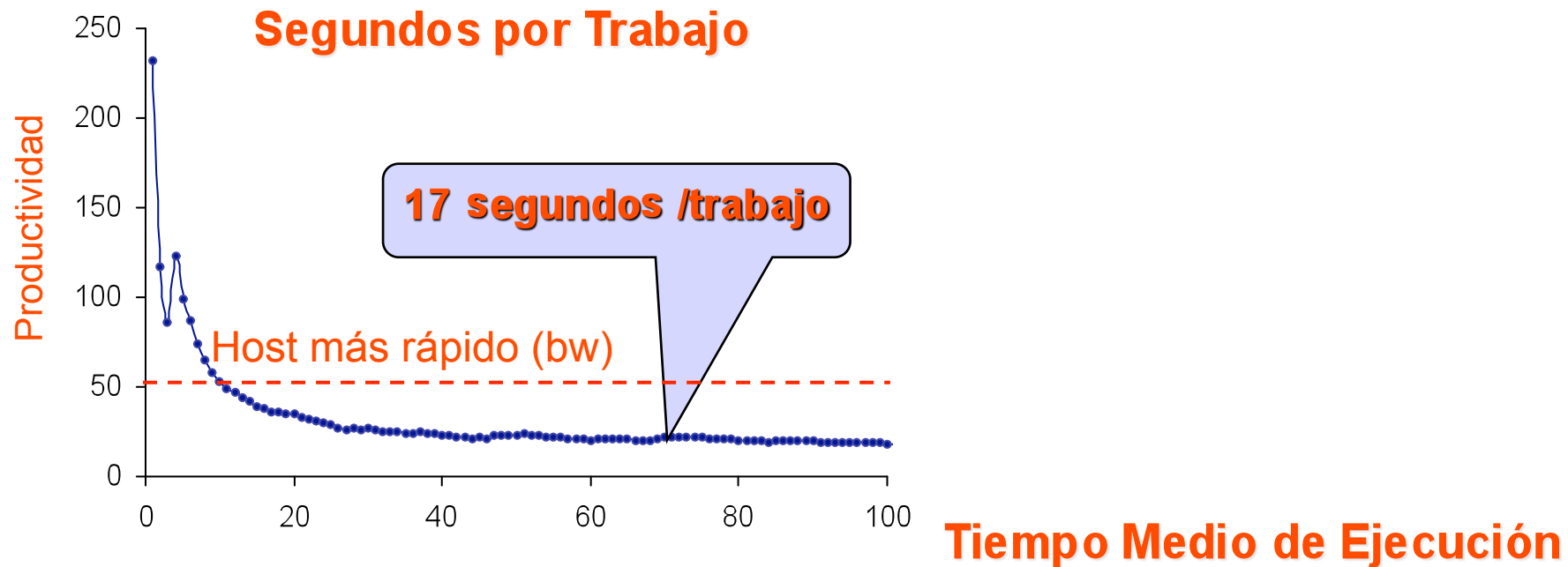
Arquitectura del Testbed

Basado en el Globus Toolkit (GT2.x), con los servicios grid básicos :

- Gestión de Recursos: **GRAM**.
- Gestión de Datos: **GridFTP**.
- Infraestructura de Seguridad : CA DataGrid-España, y CA's existentes.
- Servicio de Información: **MDS**, únicamente GRIS local de los recursos.

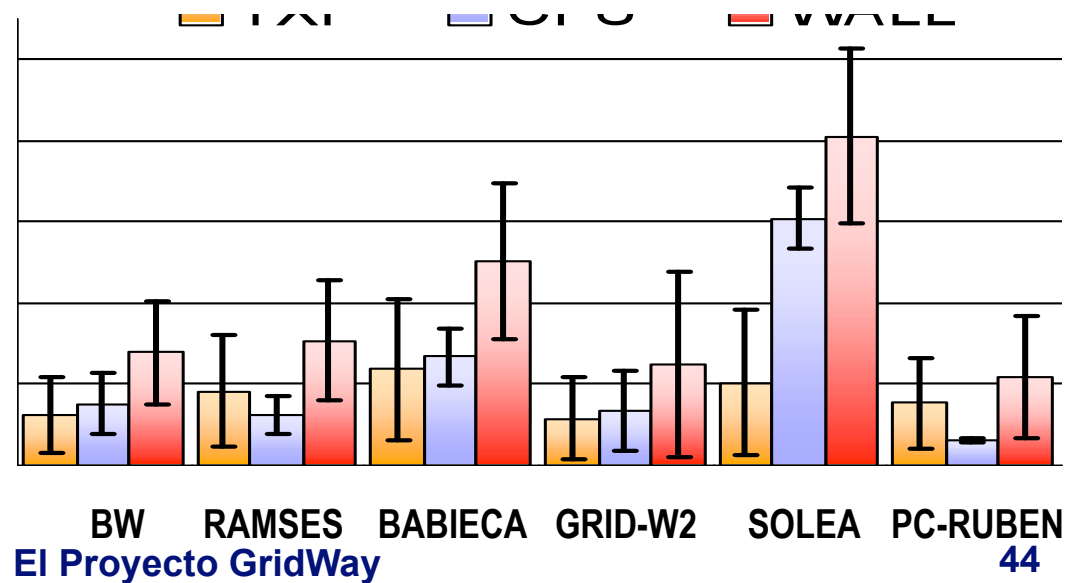
Nombre	V.O.	Modelo	CPUs	Rendimiento	S.O.	GRAM
bw	CESGA	Intel PIII	16	16 GFLOPS	Linux 2.4	PBS
grid-w2	PIC-IFAE	Intel P4	4	8 GFLOPS	Linux 2.2	PBS
ramses	DSIC-UPV	Intel PIII	20	17 GFLOPS	Linux 2.4	PBS
sherlock	DIF-UM	Intel PIII	1	550 MFLOPS	Linux 2.4	FORK
babieca	CAB	Compaq DS10	28	30 GFLOPS	Linux 2.2	PBS
pc-ruben	DACYA-UCM	Intel P4	1	2.4 GFLOPS	Linux 2.4	FORK
solea	QUIM-UCM	Sun Entr. 250	2	1.2 GFLOPS	Solaris 8	FORK
TOTAL			71	75 GFLOPS		

4. El Agente de Gestión de Trabajos - Resultados más Relevantes



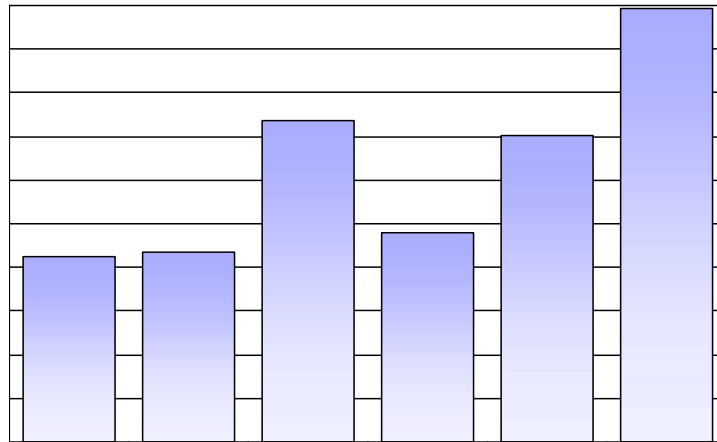
- Incluye la sobrecarga introducida por Globus
- El Grid es un entorno dinámico

Tiempo (Segundos)

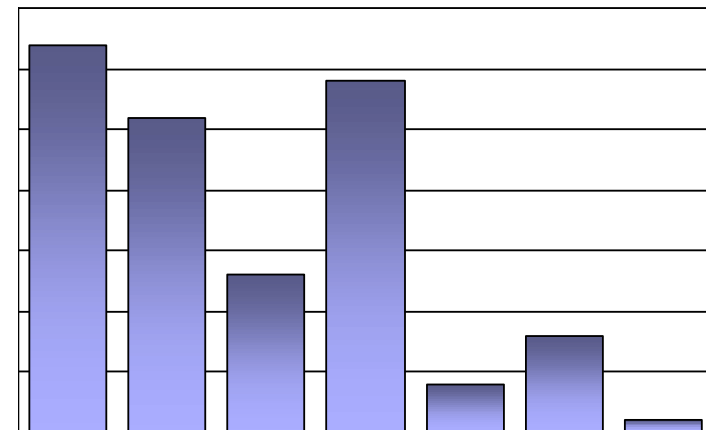


4. El Agente de Gestión de Trabajos - Resultados más Relevantes

Ganancia por V.O.



Planificación GridWay



Número de Trabajos

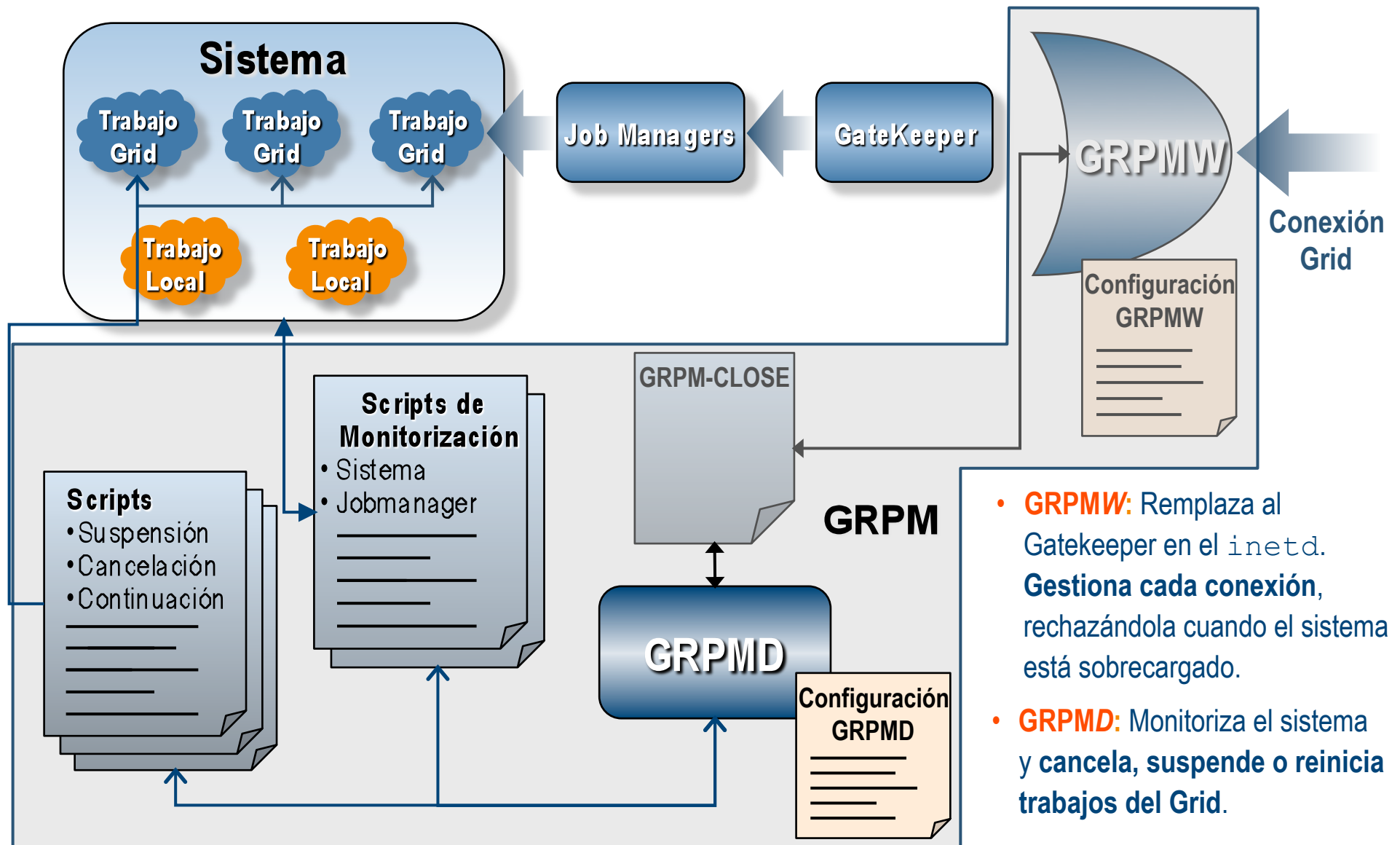
V.O	TIEMPO
CESGA	58 min.
UPV	59 min.
CAB	126 min.
IFAE	63 min.
UCM	110 min.
UM	4 días

**Tiempo de
Simulación
30 Minutos**

Contenidos

1. Visión Global de la Tecnología Grid
2. El Grid como Entorno Altamente Dinámico
3. El Proyecto GridWay
4. El Agente de Gestión de Trabajos
- 5. El Gestor de Rendimiento**
6. Comparación con Proyectos Semejantes
7. Trabajo Futuro

5. El Gestor de Rendimiento - Arquitectura



5. El Gestor de Rendimiento - Resultados más Relevantes

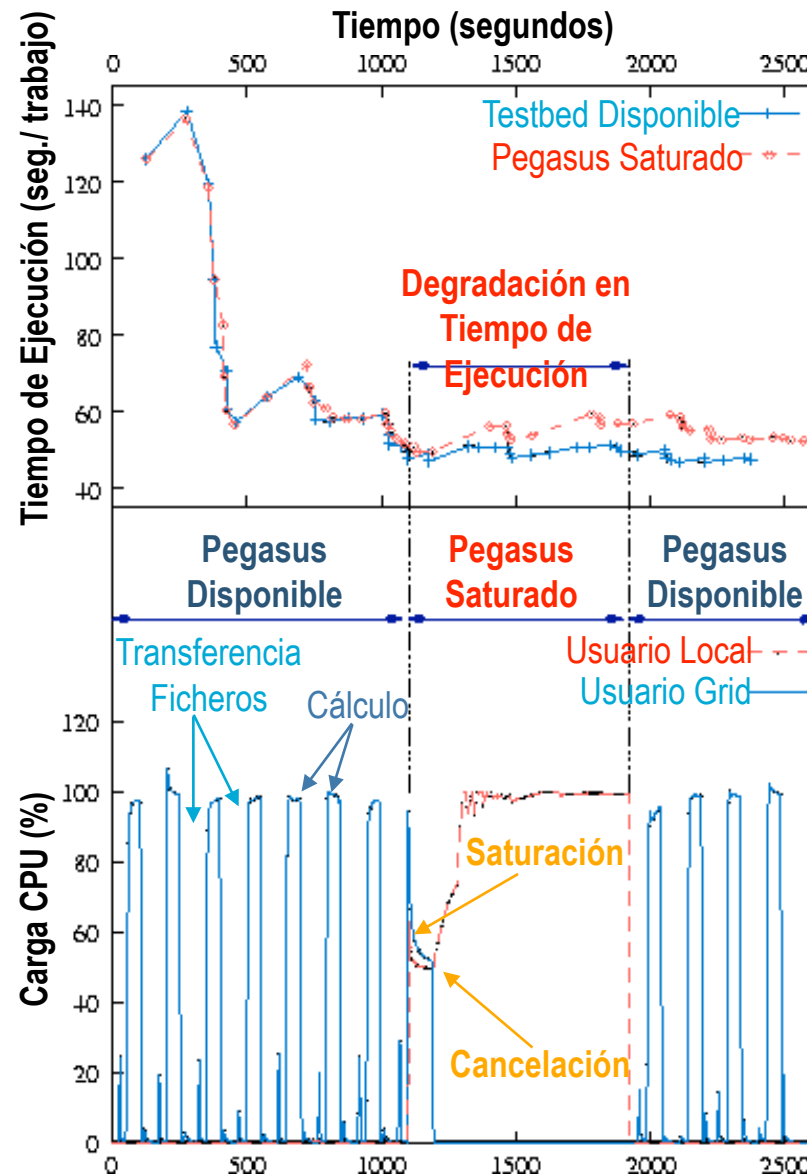
Descripción del CAB/UCM Grid:

Host	Modelo	Hz	S.O.	Memoria	Nodos	GRAM
ursa	Sun Blade 100	500Mhz	Solaris 8	256MB	1	fork
draco	Sun Ultra 1	167Mhz	Solaris 8	128MB	1	fork
pegasus	Intel Pentium 4	2.4 Ghz	Linux 2.4	1GB	1	fork
babieca	Alpha DS10	466 Mhz	Linux 2.4	1GB	4	pbs
solea	Sun Enterprise 250	296 Mhz	Solaris 8	256MB	2	fork

Descripción del Experimento:

- Mantener el Rendimiento de una estación (**pegasus**) según:
 - `MAX_GRID_USERS = 1`
 - `CANCEL = local_cpu_load > 25`
 - `RESUME = local_cpu_load < 15`
- Ejecución de una aplicación de Alta Productividad (CFD $Re=10^2 - 10^4$)

5. El Gestor de Rendimiento - Resultados más Relevantes



Contenidos

1. Visión Global de la Tecnología Grid
2. El Grid como Entorno Altamente Dinámico
3. El Proyecto GridWay
4. El Agente de Gestión de Trabajos
5. El Gestor de Rendimiento
- 6. Comparación con Proyectos Semejantes**
7. Trabajo Futuro

6. Comparación con Proyectos de Investigación Semejantes

- **Investigación en Planificación**

- Nimrod/G, Condor/G, Apples y EveryWare
- **Ninguna presenta una ejecución auto-adaptativa**

- **Ejecución Auto-Adaptativa**

- Proyecto GrADS
- **Objetivos similares:** simplificar computación distribuida en entornos heterogéneos
- **Ámbito diferente:** el entorno de ejecución *GridWay* consiste en un agente que incluye los mecanismos necesarios para la ejecución transparente y adaptativa de trabajos en Grids

- **Ventajas en General:**

- **No está limitada a una clase específica** de aplicaciones
- **No requiere la instalación de nuevos servicios**
- **No requiere modificar el código**
- **Es extensible y adaptable** a los servicios de un Grid determinado

- **No existen herramientas de planificación Grid con características semejantes**

Contenidos

1. Visión Global de la Tecnología Grid
2. El Grid como Entorno Altamente Dinámico
3. El Proyecto GridWay
4. El Agente de Gestión de Trabajos
5. El Gestor de Rendimiento
6. Comparación con Proyectos Semejantes
- 7. Trabajo Futuro**

7. Trabajo Futuro

1. Continuación de la Investigación en Grids Computacionales

- Investigación sobre **tolerancia a fallos correspondiente al cliente**
- Investigación sobre **calidad de servicio**
- Desarrollo de un API y un conjunto de comandos para especificación de problemas en terminología Grid

2. Integración en Grid de Datos

- Investigación sobre planificación en función de potencia de cálculo de los recursos y distancia del nodo de ejecución a las diferentes **replicas** y su tamaño
- Desarrollo de **selector de recursos de almacenamiento**, que considere la proximidad a un conjunto de réplicas de ficheros pertenecientes a una colección lógica.

3. Conversión a Servicio Grid

- Implementación de la herramienta como **servicio Grid**

MUCHAS GRACIAS POR SU ATENCIÓN

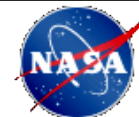


ASDS-DACYA-UCM



RedIRIS

RedIRIS



ICASE-NASA Langley



CAB-CSIC/INTA